Mean Field Approximations to Hopfield Nets and Boltzmann Machines

15-486/782: Artificial Neural Networks
David S. Touretzky

Fall 2006

1

The Basic Idea

- Replace binary units (0/1 or +1/-1) with continuous-valued units.
- State space:
  - was: “corners of a hypercube” (discrete)
  - now: “interior of the hypercube” (continuous)
- Allows a Hopfield net to consider multiple solutions simultaneously as mixtures of binary states.
- Allows a Boltzmann Machine to estimate $<S_i>$.

2

Generalized Hopfield Model (Binary Units, Stochastic Update)

V_i=*state*
I_i=external input
U_i=*threshold*
T_{ij}=coupling matrix, $T_{ij}=T_{ji}$, $T_{ii}=0$

$$V_i = sgn\left(\sum_j T_{ij} V_j + I_i - U_i\right)$$

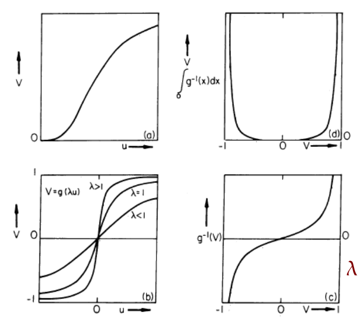
$$E = -\frac{1}{2}\sum_{i \neq j} T_{ij} V_i V_j - \sum_i I_i V_i + \sum_i U_i V_i$$

3

Continuous, Deterministic Version of Generalized Hopfield Model

Nonlinear units: $V_i = g(\lambda\, u_i)$

tanh — scale factor — net input

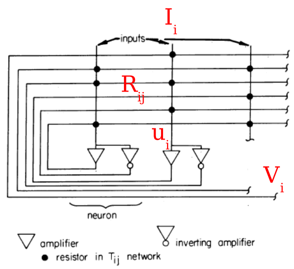


$$\lambda\, u_i = g^{-1}(V_i)$$

4

RC Circuit

$$
\begin{aligned}
R_{ij} &= 1/|T_{ij}| && \text{synapse conductance} \\
\rho_i &= \text{input resistance of amplifier i} \\
1/R_i &= 1/\rho_i + \sum_j 1/R_{ij} && \text{input resistance of neuron i} \\
u_i &= g^{-1}(V_i) \\
C_i \frac{du_i}{dt} &= \sum_j T_{ij} V_j - u_i/R_i + I_i
\end{aligned}
$$



Each neuron is a pair of amplifiers.

Separate outputs V_i and $\bar{V}_i$ for + and - weights.

Energy Function

$$E = -\frac{1}{2}\sum_i \sum_j T_{ij} V_i V_j + \sum_i (1/R_i)\int_0^{V_i} g_i^{-1}(V)\,dV + \sum_i I_i V_i$$

$$\frac{dE}{dt} = -\sum_i dV_i/dt \underbrace{\left(\sum_j T_{ij} V_j - u_i/R_i + I_i\right)}_{C_i(du_i/dt)}$$

$$
\begin{aligned}
\frac{dE}{dt} &= -\sum C_i (dV_i/dt)(du_i/dt) \\
&= -\sum C_i g_i^{-1'}(V_i)(dV_i/dt)^2
\end{aligned}
$$

Lyapunov Function ⇒ Convergence

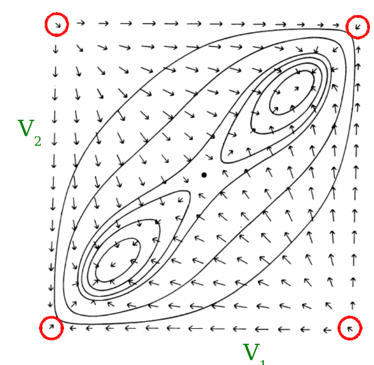
$$\frac{dE}{dt} = -\sum C_i g_i^{-1'}(V_i)(dV_i/dt)^2$$

Since $g^{-1}(V_i)$ is monotone increasing and C_i is positive, $dE/dt \leqslant 0$, and the system will settle into a stable state.

For small $T=1/\lambda$, the stable states are hypercube corners.
For large T (small λ), the stable states are interior points.

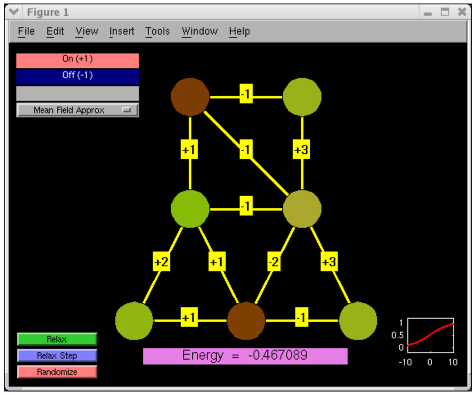
Stable States of a Two-Neuron System for Moderate λ

$$
\begin{aligned}
T_{12} &= T_{21} = 1 \\
\lambda &= 1.4 \\
g(u) &= (2/\pi)\tan^{-1}(\pi\lambda u/2)
\end{aligned}
$$



Note the stable states are interior to the corners.

MATLAB Simulation

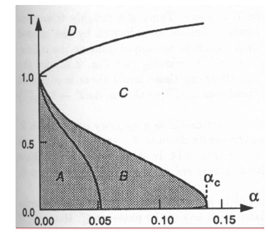


Mean Field Approximation to the Stochastic Hopfield Model

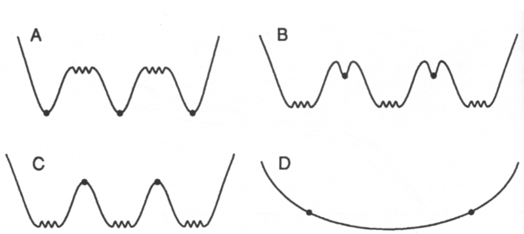
Replace V_i by $\langle V_i \rangle$
Or calculate directly as: $V_i = \tanh(\lambda u_i)$

Temperature $T = 1/\lambda$
Loading factor $\alpha = P/N$

A: memory states
B: spin glass states dominate
C: spurious states dominate
D: $\langle V_i \rangle = 0$



Energy Landscape at Different Temperatures



A: memory states are the principal minima
B: spin glass states dominate: linear combinations of odd numbers of memory states
C: spurious states dominate
D: $\langle V_i \rangle = 0$

Conclusions of Hopfield (1984)

You can build “neuron-like” circuits using amplifiers and RC networks.

Analog values correspond to:

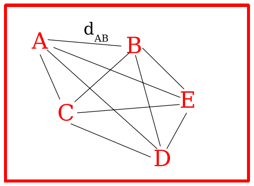
- membrane voltage?
- spike rate?

Overall behavior of these networks is similar to stochastic binary networks, even though the components are analog devices and even noisy.

Hopfield & Tank: Neural Computation of Decisions in Optimization Problems

Let's solve the Traveling Salesman Problem.

Cities



Given a table of inter-city distances d_{XY} find a minimal-length tour.

X,Y = (rows); i,j = (columns)

	1	2	3	4	5
A	0	1	0	0	0
B	0	0	1	0	0
C	1	0	0	0	0
D	0	0	0	0	1
E	0	0	0	1	0

A single "1" in each row and each column.

How to Solve TSP

Express TSP as a set of constraints on states of the N^2 binary units in the N×N solution matrix.

Map the constraints into terms in an energy function.

Find the minimum energy state.

Read off the solution to the original TSP problem.

We'll use 0/1 units for this task.

Constraint #1: No City Visited Twice

$$E_1 = \frac{A}{2} \sum_X \sum_{i \neq j} V_{X,i} V_{X,j}$$

X ranges over cities. All terms of the summation will be zero if each row contains at most one 1.

Constraint #2: Can't Be Two Places At Once

$$E_2 = \frac{B}{2} \sum_{X \neq Y} \sum_i V_{X,i} V_{Y,i}$$

i ranges over tour positions. All terms of the summation will be zero if each column contains at most one 1.

Constraint #3: A Total of *n* Cities Is Visited

$$E_3 = \frac{C}{2}\left[\left(\sum_{X,i} V_{X,i}\right) - n\right]^2$$

This value will be zero if there are exactly n ones in the matrix.

Constraint #4: Path Cost Determined by d_{XY}

$$E_4 = \frac{D}{2}\sum_{X \neq Y}\sum_i d_{XY} \cdot V_{X,i} \cdot \left(V_{Y,i-1} + V_{Y,i+1}\right)$$

If we visit X at step i, and Y at step i-1 or i+1, then count the distance d_{XY} as part of the tour.

Complete Energy Function

$$E = E_1 + E_2 + E_3 + E_4$$

The constants A, B, C, and D determine the relative weight of each constraint.

How can we derive the weights T_{ij}?

$$E = = \frac{1}{2}\sum_{i,j} T_{ij} V_i V_j$$

so $$T_{ij} = -\frac{\partial^2 E}{\partial V_i \partial V_j} = \frac{\partial u_i}{\partial V_j}$$

Weights for TSP

$$T_{Xi,Yj} = \frac{\partial^2 E}{\partial V_{X,i} \partial V_{Y,j}}$$

$$T_{Xi,Yj} = -A\,\delta_{XY}\left(1-\delta_{ij}\right) - B\,\delta_{ij}\left(1-\delta_{XY}\right) - C - D \cdot d_{XY}\left(\delta_{j,i-1} + \delta_{j,i+1}\right)$$

where $\delta_{ab} = \begin{cases} 1 & \text{if a=b} \\ 0 & \textit{otherwise} \end{cases}$

Solve TSP by using a continuous-valued Hopfield net and gradually lowering T.

N cities, N^2 units, N^4 weights.

Simulating TSP

$$\frac{du_{X,i}}{dt} = -u_{Xi}/\tau - A\sum_{j\neq i} V_{Xj} - B\sum_{Y\neq X} V_{Yi} - C\left[\left(\sum_X \sum_j V_{Xj}\right) - n\right] - D\sum_Y d_{XY}\left(V_{Y,i-1} + V_{Y,i+1}\right)$$

$$V_{X,i} = \frac{1}{2}\left(1 + \tanh\left(u_{X,i}/u_0\right)\right)$$

$u_0 = 1/\lambda$: gain parameter

Parameter Values for 10 City Problem

A = 500
B = 500
C = 200

A, B, C large: focus on legal tours

D = 500

D large: minimize path length

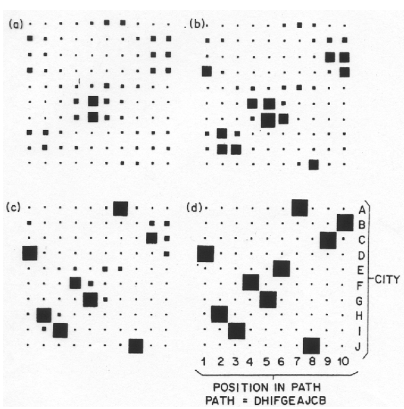
n = 15 (heuristic for 10 cities)

u_{00}=value that gives $\sum_X \sum_i V_{X,i} = 10$

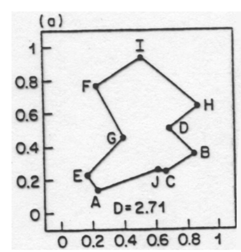
Initial $u_i = u_{00} + \delta u$

Random $\delta u \in [-0.1, +0.1]$

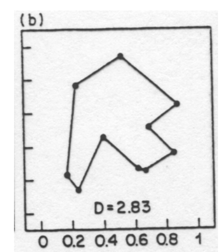
Sample Run on 10 Cities



Simulation Results



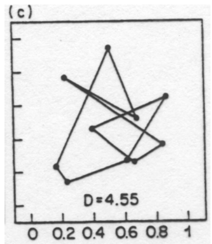
Optimal solution



Pretty good solution

Good results with gain $u_0 = 0.02$ (i.e., $\lambda = 50$)
Better results if u_0 was slowly decreased.

A Stochastic, Discrete (0/1) Network Produced Random Results



Is This a Good Way to Solve TSP?

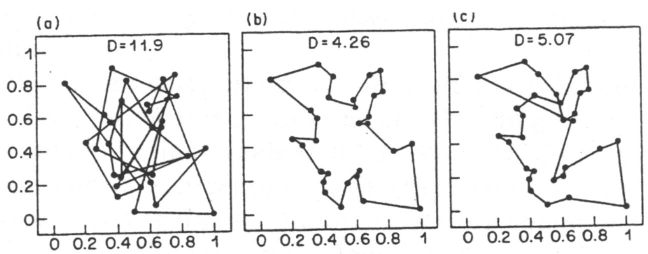
Not all stable states were valid solutions. (16/20)

Not all solutions were optimal. (50%, for 10 cities.)

There are non-neural net algorithms that give better results.

But the basic idea of encoding a problem in an energy function is very interesting, and suggestive of general principles for how brains may compute.

Needs Further Tuning for 30 City Problem



(a) Random tour; (b) Lin-Kernighan (non-neural) algorithm; (c) analog network with slowly decreasing gain.

Mean Field Boltzmann Machines

Same idea as for Hopfield nets:

- Continuous-valued units
- Instead of annealing (stochastic units and falling temperature), use continuous units and increasing gain.
- Random initial conditions to break symmetry.
- Wake/sleep learning algorithm.

Settling is much faster than annealing.

Learning is still slow, but not as bad as with annealing.

Potential drawback: local minima.